EXERCISE 1

Question 1 **. What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection? What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?**

The IP address of gaia.cs.umass.edu is 128.119 .245.12. It is sending and receiving TCP segments from port number 80. The IP address of the client is 192.168.1.102 and the TCP port number is 1161.

Question 2. **What is the sequence number of the TCP segment containing the HTTP POST command? Note that in order to find the POST command, you'll need to dig into the packet content field at the bottom of the Wireshark window, looking for a segment with a "POST" within its DATA field.**

The sequence number of the TCP segment is 232129013.

Question 3. **Consider the TCP segment containing the HTTP POST as the first segment in the TCP connection. What are the sequence numbers of the first six segments in the TCP connection (including the segment containing the HTTP POST) sent from the client to the web server (Do not consider the ACKs received from the server as part of these six segments)? At what time was each segment sent? When was the ACK for each segment received? Given the difference between when each TCP segment was sent, and when its acknowledgement was received, what is the RTT value for each of the six segments? What is the *EstimatedRTT* value (see relevant parts of Section 3.5 or lecture slides) after the receipt of each ACK? Assume that the initial value of *EstimatedRTT* is equal to the measured RTT (*SampleRTT*) for the first segment, and then is computed using the *EstimatedRTT* equation for all subsequent segments. Set alpha to 0.125.**

Packet No.	Sequence No.	Time Sent (s)	Time ACK Received	RTT (s)	Estimated RTT (s)
4	232129013	0.026477	0.053937 (pkt 6)	0.027460	0.027460
5	232129578	0.041737	0.077294 (pkt 9)	0.035557	0.028472
7	232131038	0.054026	0.124085 (pkt 12)	0.070059	0.03367
8	232132498	0.054690	0.169118 (pkt 14)	0.114428	0.043765
10	232133958	0.077405	0.217299 (pkt 15)	0.139894	0.055781
11	232135418	0.078157	0.267802 (pkt 16)	0.189645	0.072514

Note: Wireshark has a nice feature that allows you to plot the RTT for each of the TCP segments sent. Select a TCP segment in the "listing of captured packets" window that is being sent from the client to the gaia.cs.umass.edu server. Then select: *Statistics->TCP Stream Graph>Round Trip Time Graph* . However, do not use this graph to answer the above question.

Question 4. **What is the length of each of the first six TCP segments?**

Packet No	Length (Bytes)
4	565
5	1460
7	1460
8	1460
10	1460
11	1460

Question 5. **What is the minimum amount of available buffer space advertised at the receiver for the entire trace? Does the lack of receiver buffer space ever throttle the sender?**

5840 bytes. No as the packet lengths of each of the sent segments never exceeded that amount. In addition to this the receiver buffer space adjusts itself (grows or decreases) depending on the network congestion.

Question 6. Are there any retransmitted segments in the trace file? What did you check for (in the trace) in order to answer this question?

No there are not any retransmitted segments in the trace file. To check for this I used the TCP time sequence stream sequence graph. As the sequence number grew consistently with time, this highlights that no packets were retransmitted. If a packet was retransmitted, the graph would have displayed a drop in the sequence number in respect to time.

Question 7. **How much data does the receiver typically acknowledge in an ACK? Can you identify cases where the receiver is ACKing every other received segment (recall the discussion about delayed acks from the lecture notes or Section 3.5 of the text).**

The receiver typically acknowledges 1460 bytes of data sent by the client. A case where the receiver is ACKing every other received segment is #60. This is shown as it returns an ACK number that is 1460 * 2 = 2920 greater than the previously sent ACK.

Question 8. **What is the throughput (bytes transferred per unit time) for the TCP connection? Explain how you calculated this value.**

Throughput is calculated by the total data sent divided by the total amount of time. Total data is the packet length not including headers. As such the throughput is:

Bytes sent from client **Bytes sent from server** **SYN Bits** **Total Time**

(232293103 – 232129012) + (883062516 - 883061785) – 2 / 5.651141

= 164820 / 5.651141

= 29166.15 Bytes/Second

~= 29 KB/s

No	Source IP	Destination IP	Protocol	Info
295	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [SYN] Seq=2818463618 win=8192 MSS=1460
296	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [SYN, ACK] Seq=1247095790 Ack=2818463619 win=262144 MSS=1460
297	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463619 Ack=1247095791 win=65535
298	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [PSH, ACK] Seq=2818463619 Ack=1247095791 win=65535
301	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [ACK] Seq=1247095791 Ack=2818463652 win=262096
302	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [PSH, ACK] Seq=1247095791 Ack=2818463652 win=262144
303	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463652 Ack=1247095831 win=65535
304	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [FIN, ACK] Seq=2818463652 Ack=1247095831 win=65535
305	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [FIN, ACK] Seq=1247095831 Ack=2818463652 win=262144
306	10.9.16.201	10.99.6.175	TCP	50045 > 5000 [ACK] Seq=2818463652 Ack=1247095832 win=65535
308	10.99.6.175	10.9.16.201	TCP	5000 > 50045 [ACK] Seq=1247095831 Ack=2818463653 win=262144

EXERCISE 2

Question 1 **. What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and server?**

The sequence number is 2818463618

Question 2. **What is the sequence number of the SYNACK segment sent by the server to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? How did the server determine that value?**

The sequence number of the SYNACK segment is 1247095790. The value of the ACK field is 2818463619. The server determines that value as it is the sequence number of the SYN segment + 1 bit (for the SYN bit).

Question 3 **. What is the sequence number of the ACK segment sent by the client computer in response to the SYNACK? What is the value of the Acknowledgment field in this ACK segment? Does this segment contain any data?**

The sequence number of the ACK segment in response to the SYNACK is 2818463619. The value of the ACK field is 1247095791. This segment does not contain any data however the SYN header value is set to 1.

Question 4 **. Who has done the active close? client or the server? how you have determined this? What type of closure has been performed? 3 Segment (FIN/FINACK/ACK), 4 Segment (FIN/ACK/FIN/ACK) or Simultaneous close?**

A simultaneous close has been performed, therefore both the client and server has done an active close. By the process of elimination, it is not a 3 segment close as 4 segments were sent. In addition to this it is not a 4 segment close as the order of the segments sent were (FIN, FIN, ACK, ACK) thus it is a simultaneous close.

Question 5 **. How many data bytes have been transferred from the client to the server and from the server to the client during the whole duration of the connection? What relationship does this have with the Initial Sequence Number and the final ACK received from the other side?**

The total data transferred from the client is 2818463653 (Final ACK) – 2818463618 (ISN) – 2 (FIN & SYN bit) = 33 bytes.

The total data transferred from the server to the client is 1247095831 (Final ACK) – 1247095790 (ISN) – 2 (FIN & SYN bit) = 39 bytes.